

**IN THE UNITED STATES PATENT AND TRADEMARK OFFICE**

Application Serial No. ....10/693,409  
Filing Date .....October 24, 2003  
Confirmation No.....1913  
Inventorship.....Snover et al.  
Applicant.....Microsoft Corp.  
Group Art Unit .....2193  
Examiner .....Wood, William H.  
Attorney's Docket No. ....MS1-1739US  
Title: Mechanism for Providing Extended Functionality to Command Line  
Instructions

**APPEAL BRIEF**

To: Commissioner for Patents  
PO Box 1450  
Alexandria, Virginia 22313-1450

From: Robert G. Hartman (Tel. 509-324-9256 x265; Fax 509-323-8979)  
Customer No. 22801

Pursuant to 37 C.F.R. §41.37, Applicant hereby submits an appeal brief for application 10/693,409, filed October 24, 2003, within the requisite time from the date of filing the Notice of Appeal. Accordingly, Applicant appeals to the Board of Patent Appeals and Interferences seeking review of the Examiner's rejections.

<b><u>Appeal Brief Items</u></b>	<b><u>Page</u></b>
(1) Real Party in Interest	3
(2) Related Appeals and Interferences	3
(3) Status of Claims	3
(4) Status of Amendments	3
(5) Summary of Claimed Subject Matter	4
(6) Grounds of Rejection to be Reviewed on Appeal	5
(7) Argument	6
(8) Appendix of Appealed Claims	23
(9) Evidence appendix	28
(10) Related Proceedings appendix	31

**(1) Real Party in Interest**

The real party in interest is Microsoft Corporation, the assignee of all right, title and interest in and to the subject Application.

**(2) Related Appeals and Interferences**

Appellant is not aware of any other appeals, interferences, or judicial proceedings which will directly affect, be directly affected by, or otherwise have a bearing on the Board's decision to this pending appeal.

**(3) Status of Claims**

Claims 1-20 stand rejected, are pending in the Application, and are set forth in the Appendix of Appealed Claims on page 23.

**(4) Status of Amendments**

A first Office Action was issued on February 7, 2006.

A Response was filed on May 5, 2006. Claims 1, 8, 9, and 16 were amended.

A Final Office Action was issued on July 24, 2006.

A Response was filed on September 29, 2006. No claims were amended.

An Advisory Action was issued on November 2, 2006.

A Notice of Appeal was filed on November 24, 2006.

**(5) Summary of Claimed Subject Matter**

A concise explanation of each of the independent claims is included in this Summary section, including specific reference characters, if any. These specific reference characters are examples of particular elements of the drawings for certain embodiments of the claimed subject matter and the claims are not limited to solely the elements corresponding to these reference characters.

With regards to claim 1, in a command line operating environment, a computer-executable method comprises executing each command on a command line in a first execution mode (Fig. 17, Block 1706; page 55, line 19 through page 56, line 4) or in an alternate execution mode (Fig. 17, Blocks 1710-1712; page 55, line 19 through page 56, line 4; page 56, line 17 through page 60, line 16), wherein executing the command in the alternate execution mode occurs when the command includes an instruction to execute in the alternate execution mode (Fig. 17, “Yes” branch of Block 1704; page 55, line 19 through page 56, line 4; page 56, lines 17-18), the alternate execution mode being provided by the command line operating environment (page 56, lines 1-3).

With regard to claim 9, at least one computer-readable medium has computer-executable instructions for performing a method, comprising receiving a command line operative by a command line operating environment that directs the performance of a task; determining by the command line operating environment if a parameter is present on the command line that is associated with a simulation mode (page 2, lines 1-4; page 4, 18-24; page 55, line 19 through page 56, line 3; page 57, line 18 through page 58, line 6; page 80, lines 7-9); if the parameter is present, simulating the performance of the task by the command line

operating environment (page 2, lines 1-4; page 4, 18-24; page 55, line 19 through page 56, line 3; page 57, line 18 through page 58, line 6; page 80, lines 7-9); and reporting the results of the simulation by the command line operating environment (page 2, lines 1-4; page 4, 18-24; page 55, line 19 through page 56, line 3; page 57, line 18 through page 58, line 6; page 80, lines 7-9).

With regard to claim 16, a system provides a command line operating environment, the system comprising a processor (Fig. 1, element 102; page 1, lines 15-18); and a memory, the memory being allocated for a plurality of computer-executable instructions which are loaded into the memory for execution by the processor (Fig. 1, element 104; page 1, lines 15-18), the computer-executable instructions performing a method comprising executing each command entered on a command line, wherein if the command includes an instruction to execute the command using extended functionality provided by the command line operating environment, executing the command using the extended functionality (page 4, lines 18-25; page 80, lines 2-12).

#### **(6) Grounds of Rejection to be Reviewed on Appeal**

Claims 1-3, 6, 8, 9-10, 12, and 16-18 stand rejected under 35 U.S.C. §102(b) as being anticipated by Altiris RapidInstall, version 3.0, "Release Notes" (hereinafter, "RapidInstall").

Claims 7 and 11 stand rejected under 35 U.S.C. §102(b) as anticipated by or, in the alternative, under 35 U.S.C. §103(a) as being obvious over RapidInstall.

Claims 4, 5, 13-15, and 19-20 stand rejected under 35 U.S.C. §103(a) as being obvious over RapidInstall in view of Applicant Admitted Prior Art (hereinafter, "AAPA").

**(7) Argument**

**A. The rejections under 35 U.S.C. §102(b) fail to establish that RapidInstall anticipates the claims against which it is cited.**

Applicant respectfully submits that the Office has not established that the RapidInstall reference anticipates the claims rejected under 35 U.S.C. §102(b). The discussion begins with a section entitled "The §102 Standard", which describes the standard by which claim anticipation is established. Next, a section entitled "Applicant's Disclosure" describes salient aspects of the present Application. A section entitled "The RapidInstall Reference" follows and explains the RapidInstall reference. Finally, a section entitled "The Claims" presents Applicant's reasoning as to why the Office has not established that RapidInstall anticipates the rejected claims.

**The §102 Standard**

Applicant first notes the requirements of MPEP §2131, which states that to anticipate a claim, the reference must teach "every element" of the claim. This section further states that:

"A claim is anticipated only if each and every element as set forth in the claim is found, either expressly or inherently described, in a single prior art reference." *Verdegaal Bros. v. Union Oil Co. of California*, 814 F.2d 628, 631, 2 USPQ2d 1051, 1053 (Fed. Cir. 1987).... "The identical invention must be shown in as complete detail as is contained in the ... claim."

*Richardson v. Suzuki Motor Co.*, 868 F.2d 1226, 1236, 9 USPQ2d 1913, 1920 (Fed. Cir. 1989). The elements must be arranged as required by the claim, but this is not an *ipsissimis verbis* test, i.e., identity of terminology is not required. *In re Bond*, 910 F.2d 831, 15 USPQ2d 1566 (Fed. Cir. 1990).

Accordingly, the reference must contain, within its four corners, exactly the subject matter of the claim, as it appears in the claim, in order to support a valid finding of anticipation. The absence from a cited §102 reference of *any* claimed element negates a finding of anticipation. *See, e.g., Kloster Speedsteel AB, et al. v. Crucible, Inc., et al.*, 793 F.2d 1565, 230 USPQ 81 (Fed. Cir. 1986) (emphasis added).

#### **Applicant's Disclosure**

Having discussed the standard against which the Office's rejections must be measured, Applicant now provides a brief description of the current Application. Applicant notes that while this description is provided to assist the reader in appreciating the subject matter described in Applicant's specification, it is not intended to imply or impose specific limitations with regard to Applicant's claims.

Perhaps a good place to start to gain an appreciation of the claimed subject matter is in Applicant's "Background" section. There, the "Background" section describes that in a command line environment, a command line interface allows a user to directly perform a task by entering in a command. For example, a command line interface may be invoked that provides a window that displays a prompt (e.g., "C:\>"). A user may type in any of numerous commands, such as "dir", at the prompt to perform the command. *Applicant's Specification*, Background.

With this setting in mind, the present Application provides extended functionality to command line instructions in a manner that is significantly different from the traditional mechanisms for extending functionality. For example, in traditional mechanisms, each command that desires the extended functionality must *incorporate the corresponding code into the command itself*. See, e.g., *RapidInstall infra*. The command itself must then parse the command string to determine whether a switch (e.g., verbose, whatif) was provided and execute the extended functionality accordingly. *In contrast*, mechanisms described within the present Application allow users to specify an argument within the command string in order to execute the extended functionality for a particular cmdlet, as long as the cmdlet incorporates a hook to the extended functionality. That is, an operating environment itself provides the extended functionality—and not merely the command itself. Thus, the present mechanisms *minimize the amount of code system administrators need to write*. In addition, the *extended functionality is implemented in a uniform manner*. *Applicant's Specification*, p. 74, lines 7-20. (emphasis added).

### **The RapidInstall Reference**

RapidInstall, meanwhile, describes a traditional mechanism as discussed above. Functionally, the RapidInstall application captures changes that occur on a personal computer when the user installs an application or makes configuration changes. RapidInstall then captures and builds these changes into a compressed, self-contained, and self-extracting *executable (.exe) file called a RapidInstall Package (or "RIP")*. This RIP file can then be loaded onto client personal



computers to make for easier installation of the new applications and configuration changes.

RapidInstall also discusses various command line switches (e.g., “-r”, “-cu”, “-si”) that can be executed from a command line. However, importantly, the RapidInstall program itself incorporates the code that handles these switches and functionality. One specific command line switch provided by RapidInstall is -si, which allows for a “Simulated Install”. Here, a user may simulate the installation of a built RIP file to determine if the user may experience any potential problems upon actual installation of the RIP file. Again, the user may use the “-si command line” to view this simulation. When the user runs the simulation via the -si command line, the RIP runs without actually making changes to the system, and a log flags any possible deployment problems. *RapidInstall 3.0 Release Notes*, p. 1. Again, the -si command *itself* provides this simulation functionality.

## The Claims

### Claim 1

**Claim 1** recites in a command line operating environment, a computer-executable method comprising (emphasis added):

- executing each command on a command line in a first execution mode or in an alternate execution mode, wherein executing the command in the alternate execution mode occurs when the command includes an instruction to execute in the alternate execution mode, *the alternate execution mode being provided by the command line operating environment.*

In making out a rejection of this claim, the Office merely cites to a single portion of RapidInstall, namely “page 1, section ‘Simulated Install -si command

line”. *Final Office Action mailed 07/24/2006*, p. 2-3. In the “Response to Arguments” section of the Final Action, the Office also states that “RapidInstall is a command line operating environment under the broadest reasonable interpretation (page 1, last paragraph, “-si command line”). Clearly, RapidInstall provide[s] command line functionality and is an environment/program/software, which operates. The claims do not require more. Therefore, as a command line operating environment, RapidInstall provides the above limitations in question.” *Id.* at p. 8.

Applicant respectfully disagrees and submits that RapidInstall fails to anticipate Applicant’s claim 1. Specifically, Applicant respectfully submits that RapidInstall at least fails to disclose an “alternate execution mode *being provided by the command line operating environment*”, as recited in Applicant’s claim. (emphasis added).

To understand the importance of this claim element, Applicant first directs the reader’s attention to the section entitled “Applicant’s Disclosure” above. In brief, this section states that traditional mechanisms for providing extended functionality to command line instructions require that each command that desires the extended functionality *actually incorporate the code into the command itself*, as opposed to “being provided by the command line operating environment”. As shown below, the RapidInstall reference merely incorporates its functionality into the code itself in accordance with traditional mechanisms.

In order to show that RapidInstall fails to disclose the language of Applicant’s claim, Applicant begins by examining the well-known and well-defined term “operating environment”. Applicant respectfully submits that this

term has a clear and unmistakable meaning to one skilled in the art. Specifically, the term “operating environment” refers to “*the environment in which users run programs*”. (See definition obtained from Webopedia in Evidence Appendix (9), pages 28-29). In some instances, an operating environment comprises a command line operating environment, such as MS-DOS or Unix Shell. Furthermore, an “operating environment” is often deemed a control program, which itself is defined as “a program that enhances an operating system by *creating an environment in which you can run other programs*”. (See definition obtained from Webopedia in Evidence Appendix (9), page 30).

In contrast, an executable file, such as the RIP file discussed above, is merely a program file that is ready to run in a particular environment. As such, an executable file *runs within* an operating environment—but is *not itself* an operating environment.

With this vital distinction in mind, Applicant now turns to the RapidInstall reference. As discussed above, the RapidInstall reference provides for a “Simulated Install” of an RIP file (i.e., *an executable file*). This simulation allows a user to simulate the installation of a built RIP file in order to determine if the user will experience any potential problems upon actual installation of the RIP file. To do so, the user may use the “-si command line”. *RapidInstall 3.0 Release Notes*, p. 1. Again, the Office currently cites to this “-si command line” as disclosing Applicant’s “command line operating environment”.

Importantly, however, nothing in RapidInstall relates to the providing of any operating environment. Instead, RapidInstall is silent on the matter. Applicant respectfully submits, however, that this silence is quite coherent and

understandable, as both the executable RIP file and the simulated command line (-si) execute *within* a traditional command line operating environment. That is, both the RIP file as well as its simulation *runs within and on* a traditional command line operating environment. As such, the -si command *itself* provides for the simulation, in contrast with a simulation “*being provided by the command line operating environment*”, as recited in Applicant’s claim. (emphasis added). Stated otherwise, RapidInstall actually discloses using a traditional command line environment where commands (i.e., executable files) incorporate all of the desired functionality. Applicant’s claim, meanwhile, recites that extended functionality (e.g. an alternate execution mode) is “provided by the command line operating environment” *itself*. (emphasis added).

To further highlight this vital distinction, imagine that a RapidInstall user wishes to simulate the installation of an RIP file. This user would first enter the -si command into a command line operating environment. *Based on the command itself*, RapidInstall would then simulate the results of running the RIP file. Importantly, if the user wished to view the simulation of a *second* installation of a file *unrelated* to the RIP file, the traditional command line operating environment would be unable to so provide the user. Furthermore, because the -si command of RapidInstall only relates to simulating the RIP file, RapidInstall would similarly not help the user in viewing simulated results of the second installation. As discussed at length above, this failing of RapidInstall is due to the fact that the simulation functionality is provided by the *-si command itself—and is not “being provided by the command line operating environment”*, as recited in Applicant’s claim. (emphasis added).

Applicant therefore respectfully submits that the RapidInstall reference fails to support a *prima facie* case of anticipation, as RapidInstall has not and cannot be shown to disclose “wherein executing the command in the alternate execution mode occurs when the command includes an instruction to execute in the alternate execution mode, *the alternate execution mode being provided by the command line operating environment*”, as recited in Applicant’s claim. (emphasis added).

For at least this reason, Applicant respectfully submits that this claim stands allowable.

Claims 2, 3, and 6-8

**Claims 2, 3, and 6-8** depend from claim 1 and, as such, the remarks made above in regards to claim 1 apply equally to these claims. The rejections of these claims are also improper as failing to disclose these claims’ own recited features which, in combination with those recited in claim 1, are not shown to be disclosed in the reference of record.

### Claim 9

**Claim 9** recites at least one computer-readable medium having computer-executable instructions for performing a method, comprising (emphasis added):

- receiving a command line operative by a command line operating environment that to directs the performance of a task;
- determining by the command line operating environment if a parameter is present on the command line that is associated with a simulation mode;
- if the parameter is present, *simulating the performance of the task by the command line operating environment*; and
- *reporting the results of the simulation by the command line operating environment.*

In making out a rejection of this claim, the Office states that this claim is “substantially the same” as claim 1 and, as such, is rejected “in the same manner” as claim 1. *Final Office Action mailed 07/24/2006*, p. 4. Therefore, Applicant respectfully submits that this claim stands allowable for at least the reasons discussed above in regards to claim 1. For example, Applicant respectfully submits that RapidInstall at least fails to disclose if the parameter is present, *simulating the performance of the task by the command line operating environment*”, as recited in Applicant’s claim. (emphasis added). Furthermore, Applicant respectfully submits that RapidInstall also fails to disclose “if the parameter is present, *simulating the performance of the task by the command line operating environment*”. (emphasis added).

### Claim 10-12

**Claims 10-12** depend from claim 9 and, as such, the remarks made above in regards to claim 9 apply equally to these claims. The rejections of these claims are also improper as failing to disclose these claims' own recited features which, in combination with those recited in claim 9, are not shown to be disclosed in the reference of record.

### Claim 16

**Claim 16** recites a system that provides a command line operating environment, the system comprising (emphasis added):

- a processor; and
- a memory, the memory being allocated for a plurality of computer-executable instructions which are loaded into the memory for execution by the processor, the computer-executable instructions performing a method comprising:
- executing each command entered on a command line, *wherein if the command includes an instruction to execute the command using **extended functionality provided by the command line operating environment**, executing the command using the extended functionality.*

In making out a rejection of this claim, the Office states that this claim is “substantially the same” as claim 1 and, as such, is rejected “in the same manner” as claim 1. *Final Office Action mailed 07/24/2006*, p. 4. Therefore, Applicant respectfully submits that this claim stands allowable for at least the reasons discussed above in regards to claim 1. For example, Applicant respectfully submits that RapidInstall at least fails to disclose “executing each command

entered on a command line, *wherein if the command includes an instruction to execute the command using extended functionality provided by the command line operating environment, executing the command using the extended functionality*". (emphasis added).

#### Claims 17-18

**Claims 17-18** depend from claim 16 and, as such, the remarks made above in regards to claim 16 apply equally to these claims. The rejections of these claims are also improper as failing to disclose these claims' own recited features which, in combination with those recited in claim 16, are not shown to be disclosed in the reference of record.

**B. The rejections under 35 U.S.C. §103(a) are improper, as the Office has failed to comply with 37 C.F.R. 1.104(d)(2) in response to Applicant's challenge of the Office's use of Official Notice.**

#### Claims 4, 5, 13-15, and 19-20

Claims 4, 5, 13-15, and 19-20 stand rejected under 35 U.S.C. §103(a) as being obvious over RapidInstall in view of AAPA. Applicant respectfully traverses the rejections for at least the following reasons: (1) these claims all depend from an allowable base claim for reasons discussed above, and (2) the Office has failed to comply with 37 C.F.R. 1.104(d)(2) in response to Applicant's challenge of the Office's use of Official Notice.

First, Applicant respectfully submits that each of claims 4, 5, 13-15, and 19-20 depend from one of independent claims 1, 9, or 16. As such, the remarks made above in regards to these corresponding independent claims apply equally to



these dependent claims. Furthermore, AAPA has not been shown to remedy the deficiencies in the rejections of the independent claims. The rejections of these claims are also improper as failing to disclose these claims' own recited features which, in combination with those recited in their corresponding independent claims, are not shown to be disclosed in the reference of record.

For at least this reason, Applicant respectfully submits that these claims stand allowable.

Secondly, Applicant respectfully submits that the rejections of these claims are improper by virtue of the Office's failure to comply with 37 C.F.R. 1.104(d)(2) in response to Applicant's timely and proper challenge of the Office's use of Official Notice. To make such a showing, Applicant will discuss the history of the rejections immediately below.

In a first non-final Office Action mailed February 7, 2006, the Office rejected claims 4 and 19, 5 and 20, and 13-15, respectively, in view of Official Notice, as reproduced respectively below:

Claims 4 and 19

*RapidInstall* did not explicitly state *wherein the alternate execution mode prompts for verification of executing the command before executing the command*. Official Notice is taken that it was known at the time of invention to make use of verification prompts. It would have been obvious to one of ordinary skill in the art at the time of invention to implement *RapidInstall* with a verification process as is often typical.

\* \* \*

Claims 5 and 20

**RapidInstall** did not explicitly state *wherein the alternate execution mode performs a security check to determine whether a user requesting the execution of the command has sufficient privileges to execute the command*. Official Notice is taken that it was known at the time of invention to make use of command security checking. It would have been obvious to one of ordinary skill in the art at the time of invention to implement **RapidInstall** with a verification as to user security or access level process as is often typical.

\* \* \*

Claims 13-15

**RapidInstall** did not explicitly state *wherein the task comprises a pipeline of executable commands, each executable command operating in a separate process; wherein the task comprises a pipeline of executable commands, each executable command operating in the same process; or wherein each executable command comprises an instantiated class*. Official Notice is taken that it was known at the time of invention to make use of multi-process/threaded systems; pipelining processes and objected oriented class/object technology. It would have been obvious to one of ordinary skill in the art at the time of invention to implement **RapidInstall** system with each of the above technologies to create a series of pipelined commands operating on either one process or multiple processes and using class instantiation.

*Office Action mailed 02/07/2006, p. 5-7.*

In response to the Office's use of Official Notice, Applicant properly made the following request in accordance with 37 C.F.R. 1.104(d)(2):

Because the Applicant is required to seasonably challenge statements by the Examiner that are not supported on the record, and failure to do so will be construed as an admission by Applicant that the statement is true (M.P.E.P. §2144.03), *the Examiner is hereby requested to cite a reference or to cite references in support of the recited elements. If the Examiner is unable to provide such a reference, and is relying on facts based on personal knowledge, Applicant hereby requests that such facts be set forth in an affidavit from the Examiner under 37 C.F.R. 1.104(d)(2).* Absent substantiation by the Examiner, it is respectfully requested that the rejection under 35 U.S.C. §103 be withdrawn.

*Response to Office Action*, filed 05/05/2006, p. 14 (emphasis added).

As shown above, Applicant comprehensively challenged the Office's use of Official Notice, and accordingly requested that the Office substantiate its position. In response to this request, however, the Office merely stated the following:

Applicant's traverse of the Official Notice of claims 4-5, 13-15 and 19-20 is not adequate. MPEP 2144.03 section C states, "[t]o adequately traverse such a finding, an applicant must specifically point out the supposed errors in the examiner's action, which would include stating why the noticed fact is not considered to be common knowledge or well-known in the art" (emphasis added). Applicant failed to question the Official Notice facts altogether and merely questions Official Notice itself and apparently the lack of a written reference (Response: page 14, first paragraph). Therefore, the Official Notice in question is taken to be Applicant Admitted Art.

*Final Office Action* mailed 07/24/2006, p. 8.

Applicant respectfully disagrees with the Office, and instead submits that Applicant properly challenged the Office's use of Official Notice. For support, Applicant cites to the applicable Federal Rule itself. Applicant notes that this Rule takes precedence over the Manual of Patent Examining Procedure (hereinafter, "MPEP"). With this in mind, 37 C.F.R. 1.104(d)(2) affirmatively and unambiguously states that "[w]hen a rejection in an application is based on facts within the personal knowledge of an employee of the Office, the data shall be as specific as possible, and the reference must be supported, when called for by the applicant, by the affidavit of such employee". (emphasis added). As discussed and reproduced above, Applicant did indeed clearly "call" for the reference to be supported. Applicant therefore respectfully submits that the Office improperly deemed the Office's Official Notice as admitted prior art.

For at least this reason, Applicant respectfully submits that the rejection is improper. Applicant accordingly requests that the rejection be withdrawn, and that these claims be forwarded onto issuance.

Furthermore, Applicant respectfully submits that even if, as the Office suggests, an applicant is required to "state why the noticed fact is not considered to be common knowledge" in order to properly traverse Official Notice, Applicant met such a burden. *Final Office Action mailed 07/24/2006*, p. 8 (citing MPEP §2144.03(c)). In fact, further Applicant stated the following in responding to the Office's use of Official Notice:

While the Applicant agrees with the Examiner that the RapidInstall reference is silent on the additional elements recited in these dependent claims, the *Applicant contends that the Examiner is impermissibly applying hindsight reasoning using the Applicant's application to show obviousness.*

*Response to Office Action*, filed 05/05/2006, p. 14 (emphasis added).

As emphasized above, Applicant submitted that the facts taken by the Office as known in the art are indeed not commonly known, as the Office was merely “applying hindsight reasoning using the...application” to make such a finding. *Id.* Therefore, Applicant did indeed state “why” the facts are not common knowledge, even though 37 C.F.R. §1.104(d)(2) requires no such statement.

For at least this additional reason, Applicant respectfully submits that the rejection is improper. Applicant accordingly requests that the rejection be withdrawn, and that these claims be forwarded onto issuance.

Finally, and as an aside, Applicant submits that the Office’s use of Official Notice in this case is particularly egregious and inappropriate. First, the Office’s own MPEP instructs that “[w]hile ‘official notice’ may be relied on, these circumstances *should be rare when an application is under final rejection*”. MPEP §2144.03(a) (emphasis added). Furthermore, Applicant notes the complexity of the claims currently rejected as being common knowledge. For instance, claims 13 and 14 both recite, in part, “wherein the task comprises a pipeline of executable commands”, and claim 15 recites, in part, “wherein each executable command comprises an instantiated class”. Again referring back to the Office’s own manual, the MPEP clearly states that “[i]t would not be appropriate for the Examiner to take official notice of facts without citing a prior art reference where the facts asserted to be well known are not capable of instant and unquestionable demonstration as being well-known. For example, *assertions of technical facts in the areas of esoteric technology or specific knowledge of the*

prior art must always be supported by citation to some reference work recognized as standard in the pertinent art.” *Id.* (citing *In re Ahlen*, 424 F.2d at 1091, 165 USPQ at 420-421) (emphasis added). Applicant respectfully submits that this passage applies directly to Applicant’s claims, all of which pertain to “a pipeline of executable commands” and one of which recites “wherein [an] executable command comprises an instantiated class”.

For at least this final reason, Applicant respectfully submits that the rejection is improper. Applicant accordingly requests that the rejection be withdrawn, and that these claims be forwarded onto issuance.

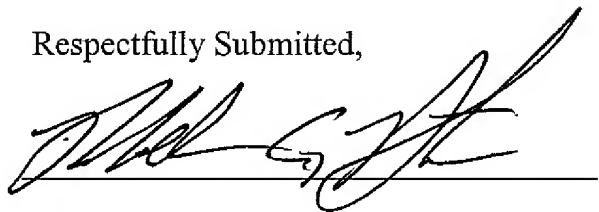
### **Conclusion**

The Office has not established a *prima facie* case of anticipation and/or obviousness for at least the reasons discussed above. Accordingly, Applicant respectfully requests that the rejections be overturned and that the pending claims be allowed to issue.

Dated: 2007-01-24

Respectfully Submitted,

By:



Robert G. Hartman  
Lee & Hayes, PLLC  
Reg. No. 58,970  
(509) 324-9256 ext. 265

**(8) Appendix of Appealed Claims**

1. (Previously Amended) In a command line operating environment, a computer-executable method comprising:

executing each command on a command line in a first execution mode or in an alternate execution mode, wherein executing the command in the alternate execution mode occurs when the command includes an instruction to execute in the alternate execution mode, the alternate execution mode being provided by the command line operating environment.

2. (Original) The computer-executable method of claim 1, wherein the alternate execution mode visually displays results of executing the command.

3. (Original) The computer-executable method of claim 1, wherein the alternate execution mode visually displays simulated results of executing the command.

4. (Original) The computer-executable method of claim 1, wherein the alternate execution mode prompts for verification of executing the command before executing the command.

5. (Original) The computer-executable method of claim 1, wherein the alternate execution mode performs a security check to determine whether a user requesting the execution of the command has sufficient privileges to execute the command.

6. (Original) The computer-executable method of claim 1, wherein executing the command in the alternate execution mode further occurs when the command line includes a switch indicating the alternate execution mode.

7. (Original) The computer-executable method of claim 6, wherein the switch comprises “whatif” and the alternate execution mode visually displays simulated results of executing the command.

8. (Previously Amended) The computer-executable method of claim 1, wherein the instruction comprises a call to a method provided by the command line operating environment.

9. (Previously Amended) At least one computer-readable medium having computer-executable instructions for performing a method, comprising:

receiving a command line operative by a command line operating environment that directs the performance of a task;

determining by the command line operating environment if a parameter is present on the command line that is associated with a simulation mode;

if the parameter is present, simulating the performance of the task by the command line operating environment; and

reporting the results of the simulation by the command line operating environment.



10. (Original) The computer-readable medium of claim 9, wherein the parameter comprises a switch.

11. (Original) The computer-readable medium of claim 10, wherein the switch comprises “whatif”.

12. (Original) The computer-readable medium of claim 9, wherein the task comprises a stand-alone executable command.

13. (Original) The computer-readable medium of claim 9, wherein the task comprises a pipeline of executable commands, each executable command operating in a separate process.

14. (Original) The computer-readable medium of claim 9, wherein the task comprises a pipeline of executable commands, each executable command operating in the same process.

15. (Original) The computer-readable medium of claim 14, wherein each executable command comprises an instantiated class.

16. (Previously Amended) A system that provides a command line operating environment, the system comprising:

a processor; and

a memory, the memory being allocated for a plurality of computer-executable instructions which are loaded into the memory for execution by the processor, the computer-executable instructions performing a method comprising:

executing each command entered on a command line, wherein if the command includes an instruction to execute the command using extended functionality provided by the command line operating environment, executing the command using the extended functionality.

17. (Original) The system of claim 16, wherein the extended functionality comprises visually displaying results of executing the command.

18. (Original) The system of claim 16, wherein the extended functionality comprises visually displaying simulated results of executing the command.

19. (Original) The system of claim 16, wherein the extended functionality comprises prompting for verification before executing the command.

20. (Original) The system of claim 16, wherein the extended functionality comprises performing a security check to determine whether a user requesting the execution of the command has sufficient privileges to execute the command.

(9) Evidence appendix: The following evidence was entered into the record by the Office in the Advisory Action dated November 2, 2006.

**InternetNewsBureau.com** Send your Press Release to More Than 14,000 Subscribing Journalists ORDER NOW!

**Internet.com** You are in the Small Business Computing Channel View Site  
**FREE IBM Online Training Tools & Resources:** Access free SSDM resource kits, IBM downloads, tools, information and tutorials for WebSphere, DB2, Rational and more.

**Internet.com (Webopedia)** The #1 online encyclopedia dedicated to computer technology  
 Enter a word for a definition... or choose a computer category.

**MENU**  
[Home](#)  
[Term of the Day](#)  
[New Terms](#)  
[Pronunciation](#)  
[New Links](#)  
[Quick Reference](#)  
[Did You Know?](#)  
[Categories](#)  
[Tech Support](#)  
[Webopedia Jobs](#)  
[About Us](#)  
[Link to Us](#)  
[Advertising](#)

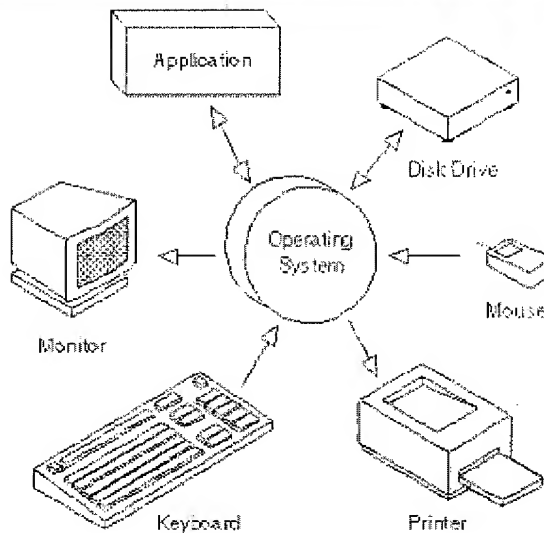
**Compare Prices:**  
   
**HardwareCentral**

**Talk To Us...**  
[Submit a URL](#)  
[Suggest a Term](#)  
[Report an Error](#)

**YAHOO! shopping**  
 Internet.com

## operating environment

Last modified: Sunday, September 01, 1996



The environment in which users run programs. For example, the DOS environment consists of all the DOS commands available to users. The Macintosh environment, on the other hand, is a graphical user interface that uses icons and menus instead of commands.

There is a thin line between operating environments and shells. Historically, shells are the interfaces to operating systems. They do not actually add any new capabilities; they simply provide a better user interface. So-called intelligent shells, however, actually extend an

**symantec. Windows Protection Solutions Center**

Secure Remote Control for IT Support Organizations  
 Remote control software is a cost-effective way of providing remote support. Some are concerned that this exposes PCs or the network to unauthorized use. This paper examines how these products provide a cost-effective help desk tool and defines security requirements.  
[Register Now to Download.](#)

Break Through the Dissimilar Hardware Restore Challenge  
 Need to minimize downtime for critical IT services by recovering entire systems to dissimilar hardware or virtual environments? Learn how the Symantec Backup Exec System Recovery can help.  
[Register Now to Download.](#)

Continuous Data Protection for Better Backup  
 Backing up mission-critical data can become a burden to IT because

- [Developer](#)
  - [International](#)
  - [Internet Lists](#)
  - [Internet News](#)
  - [Internet Resources](#)
  - [IT](#)
  - [Linux/Open Source](#)
  - [Personal Technology](#)
  - [Small Business](#)
  - [Windows Technology](#)
  - [xSP Resources](#)
  - [Search internet.com](#)
  - [Advertise](#)
  - [Corporate Info](#)
  - [Newsletters](#)
  - [Tech Jobs](#)
  - [E-mail Offers](#)
- 
- [Be a Commerce Partner](#)
  - [Online Masters](#)
  - [College Degrees Online](#)
- 
- [Cheap Plasma TVs](#)
  - [Mortgage Refinance](#)
  - [Franchise Directory](#)
  - [Server Racks](#)
  - [Televisions](#)
  - [Domain registration](#)
  - [Cheap Digital Camera](#)
  - [GPS](#)
  - [GPS](#)
  - [Desktop Computers](#)
  - [Compare Prices](#)
  - [Mp3 Player Reviews](#)

operating system's capabilities, so there is little difference between intelligent shells and operating environments.

Operating environments are sometimes called control programs.

data volumes are growing at 40-50% each year. Improve overall data protection without a costly solution that weighs down IT. Learn more. [Register Now to Download.](#)

[Optimizing Performance of the Continuous Protection Server](#)  
The stress points that continuous data protection places on system architectures are different from traditional backup and recovery technologies. Learn how one customer characterizes these points and quantifies best practices. [Register Now to Download.](#)

[Webcast: Best Practices for Protecting Microsoft Exchange with Backup Exec](#)

Learn how to manage your applications in an efficient manner for faster restores and minimized impact on business productivity.

[Register Now to Watch.](#)

[Visit the Symantec Windows Protection Solution Center](#)

[E-mail this definition to a colleague](#)

For internet.com pages about **operating environment** [CLICK HERE](#). Also check out the following links!

**LINKS**

 = Great Page!

#### **Related Categories**

[Operating Systems](#)

#### **Related Terms**

[control program](#)

[environment](#)

[graphical user interface](#)

[JVM](#)

[Microsoft Windows](#)

[operating system](#)

[shell](#)

[virtual machine](#)

**(Webopedia)**

[Give Us Your](#)

[Feedback](#)

**I) N) B) InternetNewsBureau.com** ➡ **Send your Press Release to More Than 14,000 Subscribing Journalists** ➡ **ORDER NOW!**

**internet.com** You are in the Small Business Computing Channel > View Sites **Small Business Computing**

**Tutorial: Build Web Services Using IBM Database Add-Ins for Visual Studio 2005** - Learn how to build a DB2 .NET application for Web services using the new IBM Database add-ins for Visual Studio 2005.

**internet.com** **(Webopedia)** **The #1 online encyclopedia dedicated to computer technology**

Enter a word for a definition...   ...or choose a computer category.

**MENU**

- [Home](#)
- [Term of the Day](#)
- [New Terms](#)
- [Pronunciation](#)
- [New Links](#)
- [Quick Reference](#)
- [Did You Know?](#)
- [Categories](#)
- [Tech Support](#)
- [Webopedia Jobs](#)
- [About Us](#)
- [Link to Us](#)
- [Advertising](#)

---

**Compare Prices**

**HardwareCentral**

---

**Talk To Us**

- [Submit a URL](#)
- [Suggest a Term](#)
- [Report an Error](#)

---

**BRANDERS.COM**

**Examples & Alerts**

---

**internet.com**

## control program

Last modified: Sunday, September 01, 1996

(1) A program that enhances an operating system by creating an environment in which you can run other programs. Control programs generally provide a graphical interface and enable you to run several programs at once in different windows.

Control programs are also called operating environments.

(2) Another term for operating system.

## Access Free Developer Tools

Download these IBM resources today!

### Download: DB2 Express-C 9

This production-ready version of DB2, complete with pureXML technology, is free to develop, deploy, and distribute. No limits—just data. Download Now!

### Tester Kit: Improve Test Efficiency and Accuracy

Get a collection of tools and best practices to help you avoid or detect flaws before deployment. In this kit, you'll receive demos, case studies, Webcasts, articles, tips, and more.

### Tutorial: Build Apps Using Asynchronous JavaScript with XML (Ajax)

Ajax enables a dynamic, asynchronous Web experience without the need for page refreshes. In this tutorial, you will learn to build Ajax-based Web applications—complete with real-time validation.

### Download: IBM Rational Data Architect v6.1

Download a free trial version of IBM Rational Data Architect and get to work designing relational and federated databases, understanding data assets and their relationships and streamlining database projects.

### Download: IBM Rational Web Developer for WebSphere Software

Here is an easy-to-learn IDE, less ambitious than IBM Rational App Developer for WebSphere, but perfect for developers to use to build, test, and deploy Web, Web services, and Java applications. Powered by Eclipse.

### Tutorial: The Ajax Transport Method

Discover three Ajax data transport mechanisms (XMLHttp, script tags, and frames or iframes) and their relative strengths and weaknesses. This tutorial provides code for both the server side and the client side and explains in detail to provide the techniques you need to put efficient Ajax controls anywhere you need them.

**(10) Related proceedings appendix. None**